

REMARKS

In accordance with the requirement in the office action, headings have been inserted and a new title that is descriptive of the invention to which claims are directed is submitted. The specification has been amended to cure the unclear clause on page 9, lines 18-20. Figures 1 and 2 have been amended to include the legend -- PRIOR ART --, as required in the office action.

The claims have been editorially reviewed and commas have been inserted where appropriate, under United States syntax. The claims also have been amended to assure open-ended coverage, where appropriate, and infringement of apparatus claims at the time the apparatus is sold, prior to being put into use. The word "means" has been deleted, where appropriate, to preclude any interpretation under 35 USC 112, paragraph 6. Laudatory statements in the claims have been eliminated. Claims 38-51 have been added to provide applicant with the protection to which is deemed entitled.

Applicant traverses the rejections of claims 16 and 34, respectively indirectly dependent on claims 3 and 22, under 35 USC 112, paragraph 1. Page 7, lines 1-6 and the third column of Table 2 of the specification enable one of ordinary skill in the art to make and/or use the invention defined by claims 16 and 34.

Applicant traverses the rejections of claims 9 and 28 under 35 USC 112, second paragraph. Applicant cannot agree with the assertion in the office action that claims 9 and 28 duplicate the

features of claims 6 and 23, upon which claims 9 and 28 ultimately depend. Claim 9 indicates the second information structure includes a discrete structure that receives and provides instruction data. Claim 6 is broader than claim 9 because claim 9 is limited to the discrete structure receiving and providing the instruction data. Claim 6 says nothing about a discrete structure receiving and providing instruction data. Claim 28 also indicates the second information structure includes a discrete structure for receiving the instruction data as input and for providing instruction data annotated with the resource information as output. Claim 23 merely states that the second information processing structure is arranged for generating resource information as an annotation to the page description language and and/or job control language. Hence, claim 23 is broader than claim 28 because claim 28 is limited to a discrete structure for receiving the instruction data. Based on the foregoing, claims 9 and 28 add limitations to claims 6 and 23 and, under United States practice, are not considered as duplicates. Accordingly, the rejection based on 35 USC 112, second paragraph, is erroneous.

Applicant traverses the various anticipation rejections of claims 1-4, 6, 7 and 9 (Office Action, page 7) claims 17-19 (Office Action, pages 14 and 15), claims 21-24, 26 and 28 (Office Action, pages 16-18) and claim 35 (Office Action, page 23) based on Gerlach, U.S. patent 5,469,532. In Gerlach, a computer interacts with a printer to control printer resources, such as fonts. Gerlach discloses a logical structure as shown in Figure 2, wherein a broad range of resources is held at a host computer having a large available memory and a small cache memory for holding resources present at the printer. A document for printing is examined by a resource assembler 208; see column 8, lines 27-33. Appropriate resources are collected in resource store 210; see column 8, lines 33-36. The resource assembler 208 then advises the printer as to which resources are

necessary to print the document; see column 8, lines 36-45. The printer responds to the resource assembler by downloading resources that the printer does not have in its local resource store; see column 8 lines 46-61. Resource scheduler 216 insures that downloads of resources are managed in a timely fashion and that the printer does not attempt to print a page until appropriate resources are present; see column 8, line 62-column 9, line 13.

The specification of the Gerlach reference provides a false impression of its relevance to the claims of the present application because the terms "resource" and "schedule" are used in the Gerlach reference and in the claims of the present application. However, these terms are used in entirely different ways in the Gerlach reference and in the claims of the present application. As indicated in the specification of the present application on page 3, lines 18-23, the computer processor of the present application performs scheduling of the use of resources by the printer, rather than following instructions received from a host computer in a linear manner. In contrast, in Gerlach the word "resources" is used to indicate whether a component is present or absent; an exemplary resource is a font. Gerlach does not disclose scheduling of what he terms a resource or the use of resources. It is not particularly meaningful to schedule the type of "resources" that Gerlach employs. If such resources are present, they can be used. If the resources are not present, they cannot be used. Gerlach is merely interested in ensuring that the resources are present at the printer. The "scheduler" in Gerlach does not change the operation of tasks relating to printing a page. Instead, the Gerlach scheduler merely schedules downloads and effectively introduces stalls if the downloading has not been completed and a necessary resource is not ready; see column 8, lines 2-6.

claimed?

Based on the foregoing, Gerlach does not disclose (1) scheduling printer processor resources for the different stages of printing a document in accordance with resource information, as required by claim 1, (2) a printer processor adapted to schedule resources for the different stages of printing the document in accordance with resource information, as required by claim 17, (3) a computer programmed to have a second information processing structure resource to generate resource information indicative of printer processor resources required by the printer at different stages of printing the document, as required by claims 21 and 35. Consequently, the anticipation rejections of independent claims 1, 17, 21 and 35, as well as dependent claims 2-4, 6, 7, 9, 18, 19, 22-24, 26 and 28 are incorrect. In addition, the rejections of dependent claims 5, 8, 10-16, 20, 25, 27, 29 and 30-34 on Gerlach in combination with various secondary references is incorrect because none of the secondary references cures the foregoing deficiencies in the Gerlach reference.

Applicant traverses the rejection of independent claims 36 and 37 based on Gerlach in view of Nagata, U.S. patent 5,696,984 under 35 USC 103(a). Claims 36 and 37 differ from Gerlach by requiring computer readable program code for causing a computer to generate resource information indicative of printer processor resources required by the printer at different stages of printing the document. As described previously, Gerlach does not disclose control of information indicative of printer processor resources required by the printer at different stages of printing the document. Because Nagata does not cure the foregoing deficiencies in Gerlach the rejection of claims 36 and 37 is in error.

Newly added dependent claims 38 and 40, respectively dependent on claims 1 and 17, require the scheduling step of claim 1 and the printer processor of claim 17 to provide resource scheduling by changing the order of operation of tasks relating to printing a document segment, that is specifically defined in dependent claims 39 and 41 as a page of the document. Newly added dependent claims 43 and 45 require the printer processor of claims 42 and 43 to be arranged such that the resource information and the scheduled resources, respectively, enable a change in the order of operation of tasks relating to printing a document segment that is specifically defined in dependent claims 43 and 45 as a page of the document. Newly added claims 46 and 48, respectively dependent on claims 36 and 37, require the program code for causing the computer to generate resource information to be arranged for causing the resource information to change the order of operation of the tasks relating to printing a document segment, specifically defined in claims 47 and 49 as a page of the document. Newly added claim 50 requires the scheduling step of claim 1 to include scheduling the printing of later pages prior to the execution of tasks relating to the printing of earlier pages. Newly added claim 51 requires the printer processor of claim 17 to be arranged for causing the printing of later pages prior to the execution of tasks relating to the printing of earlier pages. The foregoing structures and steps of newly added claims 38-51 are clearly not disclosed nor made obvious by the art of record.

The Abstract has been amended to conform with preferred U.S. practice.

In view of the foregoing amendments and remarks, favor reconsideration and allowance are respectfully requested and deemed in order.

To the extent necessary, a petition for an extension of time under 37 C.F.R. 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account 07-1337 and please credit any excess fees to such deposit account.

Respectfully submitted,

LOWE HAUPTMAN GILMAN & BERNER, LLP

A handwritten signature in black ink, appearing to read "Allan M. Lowe", is written over the printed name.

Allan M. Lowe
Registration No. 19,641

Customer Number: 22429
1700 Diagonal Road, Suite 300
Alexandria, Virginia 22314
(703) 684-1111
(703) 518-5499 Facsimile
Date: April 19, 2004
AML/pjc

PRINTING OF DOCUMENTS FROM A COMPUTER

TITLE OF INVENTION:

METHOD OF AND APPARATUS FOR CONTROLLING THE PRINTING OF
DOCUMENTS FROM A COMPUTER

CROSS-REFERENCE TO RELATED APPLICATIONS:

None

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT:

None

INCORPORATION-BY-REFERENCE OF MATERIAL SUBMITTED ON A COMPACT DISC:

None

Background of Invention

Field of Invention

The invention relates to an improved method of printing documents with a printer from a computer, and to apparatus and systems adapted for use of this approach. In particular, the invention relates to more efficient use of processor resources in the printing of documents.

Background Art

A conventional printing process from a computer is rendered schematically in Figure 1. The document is generated on computer 8 - this may be for example a personal computer, a workstation, or substantially any other device with a processor on which documents may be held or processed and which are connectable with a printer for producing a hard copy of the electronically held document. Typically, the document will be generated by an application 1, such as a word processing program or a spreadsheet. The document will generally be provided by the application 1 in a printer independent form 5.

This printer independent form of the document 5 is sent to a print driver 2. The print driver 2, which is typically customised to a particular model or make of printer, takes this printer independent form of a document and converts it into a printer specific page description 6 which conforms to the printer's page description language (PDL).

The printer driver 2 may also add information about the job (of printing the document), such as for example the number of copies, binding options or duplex options. This job information is provided in a job control language (JCL). In common page description languages such as Postscript (a Registered Trade Mark of Adobe Systems Incorporated), certain job information is mixed into the description of a page, as will be described further below.

The printer specific page description 6 (including any job information) is then sent to print spooler 3. The print spooler 3 is a mechanism for handling communication between the computer 8 and the printer 9 which is responsible for printing the print job. The printer specific page description is sent at the appropriate time from the print spooler 3 according to whatever protocols apply to the transport connection 7 between the computer 8 and the printer 9, and is received by the printer processor 4. The printer processor 4 observes the stream of PDL and JCL that it receives from the print spooler 3, and controls the printer to generate printed pages accordingly.

It should be noted that certain of the resources associated with the computer (in particular the print spooler 3) may not themselves be a physical part of the computer 8, but instead a part of a computer network to which both the computer 8 and the printer are attached. For convenience, throughout this specification such components associated with the computer and not specifically a part of the printer will be referred to as "of the computer".

A difficulty in conventional printing systems of this type is that the demands on the printer processor 4 can be very heavy at certain times, and very light at others. For example, consider a print job in which 1 in every 20 pages require a group of unusual fonts to be synthesised together. This synthesis results in a computational load too great for the printer processor to complete within the time normally allocated to the processing of a single page. If the fonts are not already present (for example, held in a cache on the printer) the result will be that the printer stalls (the paper feed pipeline is halted while the printer processor synthesises the necessary characters, and then restarted thereafter). Such stalls can be very expensive in resource consumption for

typical modern printers: such a pipeline may be many pages long, and a printer drum may need to be brought back up to temperature before paper can again be fed through the marking engine of the printer. Such circumstances could lead to a halving of print capacity. Situations of this general type (areas which require high consumption of resources scattered among areas which are not so computationally demanding) are common.

A mechanism for improving printer performance has been proposed by Peerless Systems (The Hard Copy Observer, September 1998, page 39) which involves deciding whether to render a page on a personal computer and then send it to the printer in TIFF format as an alternative to sending a page to the printer in the normal manner. This allows particularly complex pages to be processed by the personal computer rather than the printer, balancing the load between the two - however, this means that the overall complexity of print processing increases somewhat (with the personal computer taking on a significant additional processing load). An effective improvement to printer performance is provided, but only by shielding the printer from "difficult" pages.

It would be desirable instead to improve the use of processor resources by the printer itself to improve printer processor effectiveness and so prevent stalling and possibly also increase printing speed.

Brief Summary of the Invention

Accordingly, the invention provides a method of printing a document from a computer with a printer, the printer having a printer processor, the method comprising: generating in the computer instruction data to enable the printer to print the document; generating in the computer resource information indicative of printer processor resources required by the printer at different stages of printing the document; sending the instruction data and the resource information from the computer to the printer; scheduling printer processor resources for the different stages of printing the document in accordance with the resource information; and printing the document with the printer processor resources as scheduled.

The present inventor has appreciated that the problem of excess computational load on the printer processor can generally be removed by allowing printer processor resources to be scheduled appropriately to the demands of a specific print job. For example, in the case indicated above (of 20 normal pages followed by 1 page with complex characters), a stall could be prevented by allowing the generation of the complex characters over the course of the preceding 20 normal pages.

Preferably, the instruction data is provided as PDL and/or JCL and the resource information is provided as annotation to the page description language and/or job control language, and wherein the method comprises after the step of generating resource information the step of annotating the instruction data with the resource information.

To enable printer processor resources to be scheduled appropriately, there must be a mechanism by which the printer processor can be provided with information which can provide an indication of the printer processor resources required by the printer at future stages of printing the document. The present inventor has realised that this information is essentially available from the PDL and JCL, although it is not provided by the PDL and JCL in a form which allows the printer processor to schedule its resources in advance. It is however possible either to change the way in which the PDL and JCL are generated, or to add an additional stage at the computer (or possibly elsewhere in a computer network of which the computer is a part) in which the PDL and JCL is filtered for the information which would be needed to determine effective printer processor resource allocation, and for this filtered output to be provided to the printer processor at least in part temporally in advance of the pages to which it relates. The first of these approaches can be realised by modification of the printer driver, and the second by either modification of the print spooler or by adding an extra software package which takes the output of the printer driver, analyses it for resource information and then annotates it accordingly.

A particularly effective form of annotation is by comments in the PDL, particularly at the head of each printed page. Different approaches to this are possible, depending on desired system performance: providing all resource information on the first page header; providing resource information as it is generated on the next available page header; and preventing any resource information from appearing on the first page header (and desirably not even analysing the first page for resource information) each have their advantages, as later discussion indicates.

The invention further provides printers, computers, computer systems, and software adapted to provide elements necessary to performance of the methods indicated above.

Brief Description of the Drawing

Embodiments of the invention will now be described, by way of example, with reference to the accompanying drawings, in which:

Figure 1 shows schematically the processing of information for printing of a document by a printer from a computer in a conventional arrangement;

Figure 2 illustrates a typical arrangement of computer and printer to which the present invention is applicable;

Figure 3 shows schematically the processing of information for printing of a document by a printer according to embodiments of the present invention.

Figure 2 shows a typical arrangement for printing from a computer. Personal computer 8 has a processor 13 and a memory 12 (in both cases there may be a number of components involved - the "processor" may include a main microprocessor and coprocessors, for example, and there may be several different forms of memory). Communication is established with a printer 9 through a network connection 11. The

printer 9 also has a processor 14 for control of the printer and a memory 15 for use by the processor.

Detailed Description of the Drawing

5 Embodiments of the present invention can be employed for use with the arrangement shown in Figure 2 - the skilled man will readily understand that the present invention can also be employed in other embodiments to other forms of connection between a computer and a printer using the principles set out herein. Embodiments of a printing process according to the present invention are now described with reference to Figure
10 3.

The initial stage of the process is (typically) as shown in Figure 1: an application 1 generates a printer independent form 5 of a document (one or more pages) to be printed. This printer independent form of the document is passed to the printer driver
15 2. At this point, there is a difference from the conventional printing process. The printer driver constructs PDL and JCL to pass to the printer 9, but also while constructing the PDL identifies information that would affect how the printer 9 allocates its resources, and writes this information to a table 31. This is not a particularly onerous task, as particular PDL and JCL instructions produced by the
20 printer driver 5 in conventional languages can be identified in advance as having such resource implications - however, in conventional languages such instructions are merely embedded into the stream at the point at which their significance arises in the document, allowing no possibility of advance management of printer resources. Information that affects resources in such a way would include use of a new font (its
25 type, size, and required characters) or of bitmapped graphics (dimensions, depth, colour scheme, file type, manipulations).

The printer specific page description 6 produced by the printer driver 2 does not contain any of the resource information provided in table 31 except as PDL or JCL
30 instructions embedded within the page description in the conventional way. Resource annotation stage 32 provides a mechanism for adding this resource information into the page description in a way that will allow the printer 9 to allocate resources

effectively. An effective mechanism is to provide the resource information held in table 31 as comments (such comments are hereafter referred to as "resource declaration") in page headers of the PDL pages. The comments may relate to the page on which they are provided as headers, or, more advantageously, may relate to later pages, in which the printer processor has more time to allocate its resources. Different strategies can be used according to the desired objective. The following approaches will be advantageous in different situations.

(i) Resource declarations for the whole document are provided on the first page of the document. This provides the maximum assistance to the printer in allocating its resources, as a full timetable can be constructed before commencement of the job. Disadvantages are that the job cannot be sent to the printer until all the resource declarations have been added, and that the first page of the document may print slowly (though subsequent pages will be provided with a minimum likelihood of stalls).

(ii) Resource declarations are provided incrementally. In this case, pages are passed to the printer as soon as this is possible, with resource declarations added to the header of the first available page. Provided the analysis to achieve page declarations is sufficiently rapid, this should generally be of comparable effectiveness to method (i) (though the risk of stalls would be at least marginally greater) and printing would commence, and probably finish, more quickly.

(iii) As either method (i) or method (ii), but adapted such that either no resource declarations are provided on the first page of the document or the first page of the document is not analysed for resource declarations or (preferably) both. This maximises the speed of printing the first page of the document (an important metric for evaluation of printers), although it may increase the risk of a stall if the first or second page of the document is particularly complex.

Essentially, the resource information is provided as a separate logical channel to the print data (provided by the conventional PDL without annotations), and use of resource declarations is simply a mechanism to allow effective combination of these two logical channels on to a single transport link 7 for communication to the printer 9.

5 Methods (i) to (iii) above merely form alternative types of this basic mechanism.

Table 1 shows for one Postscript example resource information determined by print driver 2 and written to table 31 (in this case, font information), together with the main features of the PDL itself.

10

| Resource Channel | Data Channel |
|------------------------|----------------------|
| <i>page1:cmr@10.3;</i> | <i>%!PSAdobe-3.0</i> |
| <i>page1:cmi@10.4;</i> | <i>%%pages:N</i> |
| <i>page2:cmi@10.1;</i> | <i>%%EndComments</i> |
| <i>page3:cmr@10.3;</i> | <i>%%Page1</i> |
| <i>page4:cmr@10.34</i> | <i>...</i> |
| <i>...</i> | <i>%%Page2</i> |
| <i>pageN:cmr@11.3</i> | <i>...</i> |
| | <i>%%Page3</i> |
| | <i>...</i> |
| | <i>...</i> |
| | <i>%%PageN</i> |

15

Table 1

When integrated into commented PDL by the resource annotation stage, the result could be as shown in any of the three columns in Table 2, depending on the method employed.

| Initial Declarations (Method i) | Incremental Declarations (Method ii) | Fast Page One (Method iii, here as modification to Method ii) |
|--|--|--|
| <i>%!PSAdobe-3.0</i> <i>%%pages:N</i> <i>%BeginResourceDeclarations</i> <i>%Resource page1:cmr@10.3;</i> <i>%Resource page1:cmi@10.4;</i> <i>%Resource page2:cmi@10.1;</i> <i>%Resource page3:cmr@10.3;</i> <i>%Resource page4:cmr@10.34;</i> <i>%Resource ...</i> <i>%Resource pageN:cmr@11.3</i> <i>%EndResourceDeclarations</i> <i>%%EndComments</i> <i>%%Page1</i> | <i>%!PSAdobe-3.0</i> <i>%%pages:N</i> <i>%BeginResourceDeclarations</i> <i>%Resource page1:cmr@10.3;</i> <i>%Resource page1:cmi@10.4;</i> <i>%Resource page2:cmi@10.1;</i> <i>%EndResourceDeclarations</i> <i>%%EndComments</i> <i>%%Page1</i> | <i>%!PSAdobe-3.0</i> <i>%%pages:N</i> <i>%%Page1</i> |
| <i>%%Page2</i> | <i>%%Page2</i> <i>%BeginResourceDeclarations</i> <i>%Resource page3:cmr@10.3;</i> <i>%EndResourceDeclarations</i> | <i>%%Page2</i> <i>%BeginResourceDeclarations</i> <i>%Resource page2:cmi@10.1;</i> <i>%Resource page3:cmr@10.3;</i> <i>%EndResourceDeclarations</i> |
| <i>%%Page3</i> | <i>%%Page3</i> <i>%BeginResourceDeclarations</i> <i>%Resource page4:cmr@10.34;</i> <i>%Resource ...</i> <i>%Resource pageN:cmr@11.3</i> <i>%EndResourceDeclarations</i> | <i>%%Page3</i> <i>%BeginResourceDeclarations</i> <i>%Resource page4:cmr@10.34;</i> <i>%Resource ...</i> <i>%Resource pageN:cmr@11.3</i> <i>%EndResourceDeclarations</i> |
| <i>%%PageN</i> | <i>%%PageN</i> | <i>%%PageN</i> |

Table 2

The resource annotation stage 32 may be a part of the printer driver itself. Other possible locations in the pipeline are possible, and table 31 can be compiled by components other than the printer driver. The resource table 31 could for example be compiled by the application 1 (in the case of certain applications which generate the PDL directly) or by the print spooler 3, or by some resource filter sitting between the application 1 and the printer driver 2, the printer driver 2 and the print spooler 3, or the print spooler 3 and the printer 9. The resource annotation stage obviously cannot be any earlier in the pipeline than the point at which the resource table 31 is compiled, but it can sit either in the same component or at a later stage.

In the Figure 3 example, the resource annotated page description 33 is sent to the print spooler 3 and then out on to the transport connection 7 to the printer 9. The resource annotated page description is received by the printer processor 4, which is adapted to recognise which of the comments in the annotated PDL are resource declarations. The resource declarations are extracted, and processing of the appropriate tasks is scheduled so that when the page to which they relate is to be printed, ~~sufficient of the task will be~~ the task will be sufficiently completed to prevent the possibility of a stall. Although the difference to the printer processor is significant in practice, significant redesign is not required. The tasks for which resources are to be allocated are not different from those required normally - the difference is only that advance notice is given, allowing the printer processor 4 to schedule tasks earlier, and more conveniently, than is possible with conventional printing approaches

An advantage of using commented Postscript (or comments in any other language recognised by the printer) is that if the printer is not enabled to handle resource allocation according to the present invention, the print job can still be processed in the normal way - the substance of the PDL as seen by the printer is unaffected and the comments are (like other comments) simply ignored. Embodiments of the invention can thus be provided to ensure backward compatibility with existing printers.

In the embodiments described above, the two logical channels - resource channel and data channel - are combined into one unified channel of annotated data. With a printer structure adapted appropriately, these two logical channels could be kept entirely separate - the data channel provided directly to the printer in the conventional way, and the resource channel provided separately (possibly on the same physical wire, but as a different stream of packets) and received and interpreted by a specifically adapted part of the printer - possibly a separate processor in the printer, or possibly as a separate activity within the existing printer processor.

to generate resource information indicative of printer processor resources required by the printer at different stages of printing the document, and to annotate the instruction data with the resource information, thereby providing the annotated instruction data for printing by the printer.